

Remarks

The above Amendments and these Remarks are in reply to the Office Action mailed June 1, 2007, and the Examiner Interview of August 30, 2007. Applicant acknowledges with thanks Examiner Nguyen and Examiner Steelman for their assistance in granting an interview on August 30, 2007, during the course of which interview various features of the claimed embodiments were discussed, the substance of which is included herein. The interview discussed the prior art and a proposed amendment to Claim 1.

I. Summary of Examiner's Objections/Rejections

Claims 1-21 and 25 were pending in the Application prior to the Office Action mailed June 1, 2007. The Office Action rejected claims 1-21 and 25.

The Office Action stated that the Specification should be amended to capitalize JAVA and be accompanied by the generic terminology.

Claim 25 was rejected under 35 U.S.C. 112 for containing JCBC and EJBs.

Claims 1-10, 12, 14-18, and 25 were rejected under 35 U.S.C. 102(e) as being anticipated by Boehme et al. (U.S. Pat. No. 6,578,191).

Claims 1-10, 12, 14-18, and 25 were rejected under 35 U.S.C. 103(a) as being unpatentable over Boehme et al. (U.S. Pat. No. 6,578,191).

Claims 11 and 13 were rejected under 35 U.S.C. 103(a) as being unpatentable over Boehme et al. in view of Cohen et al. (U.S. Pat. No. 6,011,918).

Claims 19-21 were rejected under 35 U.S.C. 103(a) as being unpatentable over Boehme et al. in view of Stapp et al. (U.S. Publication. No. 2004/0015832).

The office action cited Foti (U.S. Pat. No. 7,181,745), Jones (U.S. Pat. No. 6,877,163), and Guthrie (U.S. Pat. No. 6,549,955), prior art made of record and not relied upon, as considered pertinent to applicant's disclosure.

II. Summary of Applicant's Response

This Request for Continued Examination cancels claims 4-9, 14-15, and 19-21; replaces claims 1-3, 12-13, 16-18, and 25; and adds new claims 26-27, leaving for the Examiner's present consideration claims 1-3, 10-13, 16-18, and 25-27. The specification was amended to correct informalities. The claims were amended to better describe embodiments of Applicant's invention. Reconsideration of the claims is requested.

III. Amendments to the Specification

The Specification was amended to replace java with JAVA™.

IV. Response to 35 U.S.C. 112 Rejection to Claim 25

Claim 25 was amended, rendering moot the 35 U.S.C. 112 rejection.

V. Response to 35 U.S.C. 102(e) Rejections to Claims 1-3, 10-12, 16-18, and 25-27

Claim 1

Claim 1 as amended states:

A high level dynamic code generation method, comprising:

- a) creating a class file container object that stores source code describing a class;
- b) adding a first source code defining a method to the class stored in the class file container object;
- c) adding a second source code into the method in the class stored in the class file container object;
- d) repeating steps b and c to populate the class stored in the class file container

object;

e) generating a tree of statements and expressions based on the class stored in the class file container object;

f) using the tree of statements and expressions to generate byte code for the class; and

g) instantiating an instance of the class.

Claim 1 defines a high level dynamic code generation method. The steps comprise: creating a class file container object that stores source code describing a class, adding a first source code defining a method to the class stored in the class file container object, adding a second source code into the method in the class stored in the class file container object, repeating the previous two steps to populate the class stored in the class file container object, generating a tree of statements and expressions based on the class stored in the class file container object, using the tree of statements and expressions to generate byte code for the class, and instantiating an instance of the class.

In the Office Action mailed June 1, 2007, Claim 1 was rejected based on Boehme. Applicant respectfully submits that Boehme's method is not a high level dynamic generation method, and consequently is a different method with different steps. Boehme's "invention does not require generation of adapter source code ..." (Boehme, col. 2, lines 65-66). Boehme's FIG. 2 demonstrates creating bytecode for the parts of a class in steps 202-208 and then assembling the bytecode together into a class in step 209. This is in contrast with Applicant's Claim 1, which creates a source code file in steps a through d, then generates a tree of statements and expressions in step e, then generates byte code in step f. Applicant respectfully submits that Boehme does not anticipate Claim 1.

Claims 2 and 3

Dependent Claim 2 requires, "wherein creating a class file container object includes selecting a class name and a super class." Dependent Claim 3 further limits Claim 2 by requiring, "the

method of Claim 2 wherein any class name and any super class can be selected.” These features describe an important difference between Claims 2 and 3 and the prior art references cited in the record. Under Claim 3, a dynamically generated class can extend any superclass. In contrast, a class generated by the prior art technology of dynamic proxies is restricted to extending specific superclasses within a framework when selecting a superclass to extend.

In the Office Action mailed June 1, 2007, Claims 2 and 3 were rejected based on Boehme. In Boehme, generating an adapter class requires extending com.ibm.bml.EventAdapterImpl (see column 4, lines 34-36). Boehme also provides that the “adapter specification will contain information necessary to …reference the adapter superclasses …” Boehme suggests that a generated adapter is based on a specific superclass, not on any superclass.

Claims 2-3, 10-12, 16-18, and 25-27

Dependent Claims 2-3, 10-12, 16-18, and 25-27 depend from Claim 1. For at least the reasons discussed above, Dependent Claims 2-3, 10-12, 16-18, and 25-27 are patentable. Dependent Claims 2-3, 10-12, 16-18, and 25-27 add their own features which render them patentable in their own right.

VI. Response to 35 U.S.C. 103(a) Rejections to Claims 1-3, 10-13, 16-18, and 25-27

Claim 1

Claim 1 as amended states:

A high level dynamic code generation method, comprising:

- a) creating a class file container object that stores source code describing a class;
- b) adding a first source code defining a method to the class stored in the class file container object;
- c) adding a second source code into the method in the class stored in the class

file container object;

- d) repeating steps b and c to populate the class stored in the class file container object;
- e) generating a tree of statements and expressions based on the class stored in the class file container object;
- f) using the tree of statements and expressions to generate byte code for the class; and
- g) instantiating an instance of the class.

Claim 1 defines a high level dynamic code generation method. The steps comprise: creating a class file container object that stores source code describing a class, adding a first source code defining a method to the class stored in the class file container object, adding a second source code into the method in the class stored in the class file container object, repeating the previous two steps to populate the class stored in the class file container object, generating a tree of statements and expressions based on the class stored in the class file container object, using the tree of statements and expressions to generate byte code for the class, and instantiating an instance of the class.

In the Office Action mailed June 1, 2007, Claim 1 was rejected based on Boehme. Applicant respectfully submits that Boehme's method is not a high level dynamic generation method, and consequently is a different method with different steps. Boehme's "invention does not require generation of adapter source code ..." (Boehme, col. 2, lines 65-66). Boehme's FIG. 2 demonstrates creating bytecode for the parts of a class in steps 202-208 and then assembling the bytecode together into a class in step 209. This is in contrast with Applicant's Claim 1, which creates a source code file in steps a through d, then generates a tree of statements and expressions in step e, then generates byte code in step f. Applicant respectfully submits that the differences between Boehme and Claim 1 are significant enough to overcome a 35 U.S.C. 103(a) rejection for obviousness.

Claims 2 and 3

Dependent Claim 2 requires, “wherein creating a class file container object includes selecting a class name and a super class.” Dependent Claim 3 further limits Claim 2 by requiring, “the method of Claim 2 wherein any class name and any super class can be selected.” These features describe an important difference between Claims 2 and 3 and the prior art references cited in the record. Under Claim 3, a dynamically generated class can extend any superclass. In contrast, a class generated by the prior art technology of dynamic proxies is restricted to extending specific superclasses within a framework when selecting a superclass to extend.

In the Office Action mailed June 1, 2007, Claims 2 and 3 were rejected based on Boehme. In Boehme, generating an adapter class requires extending com.ibm.bml.EventAdapterImpl (see column 4, lines 34-36). Boehme also provides that the “adapter specification will contain information necessary to …reference the adapter superclasses …” Boehme suggests that a generated adapter is based on a specific superclass, not on any superclass.

Claims 2-3, 10-13, 16-18, and 25-27

Dependent Claims 2-3, 10-13, 16-18, and 25-27 depend from Claim 1. For at least the reasons discussed above, Dependent Claims 2-3, 10-13, 16-18, and 25-27 are patentable. Dependent Claims 2-3, 10-13, 16-18, and 25-27 add their own features which render them patentable in their own right.

VII. Prior Art Made of Record and Considered Pertinent to Applicant’s Disclosure

The Office Action mailed June 1, 2007, cited Foti (U.S. Pat. No. 7,181,745), Jones (U.S. Pat. No. 6,877,163), and Guthrie (U.S. Pat. No. 6,549,955), prior art made of record and not relied upon,

as considered pertinent to applicant's disclosure. Foti, Jones, and Guthrie are concerned with dynamic generation of adapters and proxies. Foti, Jones, and Guthrie are closely related to Boehme, and the differences discussed above with Boehme also apply to Foti, Jones, and Guthrie.

Foti's "invention avoids compiling the adapter class by building the adapter class using bytecodes, rather than by building the adapter class in a high-level source code language." (col. 5, lines 25-28). The invention of Jones is described in FIG. 4, where in step 404-406 parts of a class are created as bytecode and then assembled into a class in step 408. Guthrie's "byte code generator 42 takes general information regarding the needed Java object and directly generates executable java code without the need for the intermediate step of creating a Java source file." (Col. 7, lines 59-63). Applicant respectfully submits that the differences discussed above between Boehme and the claims also apply between the claims and Foti, Jones, and Guthrie.

VIII. Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration is requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: August 31, 2007

By: /Thomas K. Plunkett/

Thomas K. Plunkett
Reg. No. 57,253

Customer No. 23910
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800
Facsimile: (415) 362-2928